

REMARKS

Reconsideration and allowance in view of the foregoing amendment and the following remarks are respectfully requested. Claims 1, 9, 16 and 23 are amended without prejudice or disclaimer. Claims 15 and 22 are cancelled.

Objections to the Abstract

The Office Action objects to the Abstract because it exceeds 150 words. Applicants have amended the Abstract and respectfully request withdrawal of the rejection in that it now does not exceed 150 words.

Objections to the Specification

The Office Action objects to the Specification because of informalities. Applicants have amended the specification according to the noted informalities. Applicants respectfully request withdrawal of the objection to the specification.

Objections to Claims

The Office Action objects claims 15 and 22 under 37 CFR 1.75(c) as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicants have cancelled claims 15 and 22 thus rendering this objection moot.

Rejection of Claims 1-3, 6-25 and 28-30 Under 35 U.S.C. §102(b)

The Office Action rejects claims 1-3, 6-25 and 28-30 under 35 U.S.C. §102(b) as being anticipated by Papineni et al. (U.S. Patent No. 6,246,981) ("Papineni et al."). Applicants have amended the claims and submit that Papineni et al. fail to teach each claim limitation.

We note that we have amended claim 1 to recite a top level flow controller that is a finite state model and that manages a process of controlling speech interaction between a spoken dialog system and a user. Applicants shall explain herein the difference between the top level

flow controller that is a finite state model and the portion cited in Papineni et al. which discuss a form-based dialog manager which will differ from the limitations recited in the claims.

We first turn to claim 1. Claim 1 recites a top level flow controller that is a finite state model and that manages a process of controlling speech interaction. The Office Action asserts that column 3, lines 34-42 and column 6, lines 10-14 teach this limitation. We note that column 3, lines 34-42 merely teach a dialog manager that includes means for formulating questions for one of prompting a user for needed information and clarifying information supplied by the user. We note that it is clear in the summary of the invention and throughout the detailed description of the preferred embodiments that the dialog manager disclosed in these portions of Papineni et al. are form based and utilize task-oriented forms and scripts for determining the behavior of the dialog manager. See column 3, lines 8-42. In fact, column 6, lines 19 and 20 expressly state that “the dialog manager of the present invention is not state-based.” Further, column 6, lines 33-34 teach “for the present invention, there is no explicit representation of dialog state.” Thus, Applicants respectfully submit that by requiring the top level flow controller of claim 1 to be a finite state model, Applicants have clearly distanced and distinguished their invention from the dialog manager which is the invention of Papineni et al. The “means” disclosed in Papineni et al. are limited to form-based dialog managers.

Column 6, lines 10-14 of Papineni et al. again teach that “the dialog manager of the present invention may be form-based. However, multiple forms may be used simultaneously, each corresponding to a task in a domain of tasks. The present invention’s dialog manager permits users to address any task any time.” Again, this further supports Applicants’ point that the dialog manager of Papineni et al. is expressly not a finite state model as is recited in claim 1.

The fact that the top level flow controller is a finite state model further has bearing on the remaining limitations of a reusable sub-dialog wherein the top level flow controller invokes the

reusable sub-dialog from a state to handle a specific task associated with a speech interaction and returns control to the top level flow controller after handling the specific task. This is because a reusable sub-dialog that is invoked by the top level flow controller (that is a finite state model) is simply not taught in the reference. For example, column 6, lines 10-14, discussed above, again highlights that the dialog manager is form-based and simply uses multiple forms each corresponding to a task. Because the dialog manager is form-based and expressly not state based, clearly it cannot invoke a reusable sub-dialog at a state of the top level flow controller.

Furthermore, the Office Action cites column 6, line 64 through column 7, line 5, which Applicants presume is for the purpose of asserting that the reference teaches that a reusable dialog is isolated from application dependencies. Applicants note that what is taught in this portion of the reference is that the dialog manager is “application blind”. They explain that this means that the application manager may be used to build a banking application, stock quote system, weather reporting system, etc. Each of these relate to different databases of information. Their point is that the dialog manager is not programmed with any specific knowledge about those databases and that application specific software is referred to as backend in their disclosure. However, it is clear in the context of a reusable sub-dialog that it is the reusable sub-dialog and not the dialog manager that is isolated from application dependencies. This portion of the reference mentions nothing about reusable sub-dialogs and therefore, does not disclose a reusable sub-dialog that is isolated from an application dependency. Applicants note support in the specification for the concept of an application dependency in paragraph [0067] which provides further information regarding what an application dependency may imply.

Applicants also note that column 6, lines 10-14 and lines 57-58 are cited as matching the limitation of the top level flow controller invoking the reusable sub-dialog to handle from a state a specific task associated with the speech interaction and returning control of the top level flow

controller after handling the specific task. Column 6, lines 10-14, are discussed in several places above related to how the dialog manager is limited to being form-based and simply using multiple forms each corresponding to a task. Lines 57-58 explain that “each form corresponds to a task in the application domain.” Applicants submit that this fails to teach the particular limitation of claim 1 wherein again, this limitation requires a top level flow controller to be a finite state model and which invokes the reusable sub-dialog from a state. Inasmuch as the dialog manager discloses in column 6 is form-based, Applicants submit that this limitation is not taught or suggested by the reference. In fact, Applicants respectfully submit that because the Papineni et al. invention explicitly requires its dialog manager to not be state based, this reference teaches away from the present invention.

Further, the Office Action asserts that the last limitation of claim 1, wherein the reusable dialog and the top level flow controller operate independent of the decision model, is also taught in column 6, lines 10-14, lines 23-24 and lines 40-44. These portions of the reference mention, starting at line 22, how “in the present invention, the users are allowed to shift focus anytime, by choosing not to answer the dialog manager’s questions and/or by addressing a different task altogether. Column 6, lines 40-44, discuss an aspect of the Papineni et al. invention where in the dialog manager does not assume that these are continuous tasks. To address each task until the task is finished or cancels it explicitly, the dialog manager does not impose any restrictions on the user insofar as selection of tasks is concerned. However, Applicants submit that these portions of column 6 as well as lines 1-14 discuss nothing regarding how the reusable sub-dialog and the top level flow controller operate independent of the decision model. Paragraph [0069] of the patent specification discusses examples of this limitation of claim 1. The example given at the end of this paragraph is that a sub-dialog, which is a rule-based dialog for collecting user information, can be called by a top level dialog, which is a simple RTN that is used to route a

call. In other words, examples of the decision model that are associated with the reusable dialog and the top level flow controller are a finite state model, a rule-based model, an RTN model used to route a call and so forth. These are different models. Clearly, such a concept is not taught in column 6 of the reference wherein they only teach that their dialog manager is based on a single model, which is a form based model. In other words, the dialog manager only uses a single model in terms of handling the various tasks in a spoken dialog. Thus, while the dialog manager may not assume that a user continues to address each task until a task is finished or cancels it explicitly, there is no discussion of using a different decision model between the forms and the dialog manager. They are all taught in the context of a single decision model. Accordingly, Applicants respectfully submit that this feature is not taught or suggested by Papineni et al.

Applicants finally note that the Examiner concedes that Papineni et al. fail to use the term “top flow controller” but that the reference takes the broadest reasonable interpretation of this by one of ordinary skill in the art that would include “means for formulating questions” and “means for inheriting information previously supplied” as integral parts of the dialog manager “as disclosed by Papineni”. The Office Action cites column 3, lines 34-42. Applicants respectfully traverse this broadest interpretation because, as is clearly set forth in the discussion above, the dialog manager that is “disclosed by Papineni” is a form-based dialog manager. Column 3, under the summary of the invention clearly sets forth that their dialog manager uses task-oriented forms repeatedly. Again, column 6, explains at lines 19-20 that their dialog manager of the present invention is “not state-based.” Accordingly, Applicants respectfully submit that the “means for formulating questions” and “means for inheriting information previously supplied” should not be interpreted as broadly as the Examiner states, but must be interpreted within the scope and bounds of what is taught by Papineni et al. which is amply supported to be limited to only a form-based dialog manager. In any event, Applicants have further amended the limitation

of a top level flow controller to be required to be a finite state model which is easily distinguishable from the broadest appropriate interpretation of the discussion in column 3, lines 34-42. Accordingly, Applicants respectfully submit that claim 1 is patentable and in condition for allowance. Claims 2-8 each depend from claim 1 and recite further limitations therefrom. Accordingly, Applicants submit that these claims are patentable and in condition for allowance.

Claim 9 is amended in a manner similar to claim 1 and accordingly includes several limitations which are not taught or suggested in Papineni et al. Applicants further submit that inasmuch as Papineni et al. clearly only teach a form-based dialog manager that the portions of Papineni et al. cited in the Office Action only are limited in their teachings to a form-based dialog manager, Applicants submit that while the spoken dialog is being controlled by the first level flow controller (which is a finite state model) the step of receiving context changing input associated with speech from a user that changes a dialog context is also not taught. Portions of column 7 are cited throughout the rejection of claim 9. Column 7, lines 1-25 of Papineni et al. further teach the interaction between the dialog manager and the backend. They explain that the backend software is application-specific software such as a proprietary database associated with a banking application, a stock quote system and so forth. Thus, they explain that each form in their system corresponds to a task in the application domain and each form has a backend command associated with the form. They teach about the ability to dynamically switch between a machine-initiative mode and a mixed-initiative mode. The dialog manager of the current invention can be started in either a mixed-initiative or a machine-initiative mode and may switch to a machine-initiative mode whenever it decides that the user needs more assistance. Then it can switch back to a previous mode when assistance is no longer necessary. Maintaining and modifying a list of currently admissible forms is implemented such that the dialog mode is dynamically shaped. Applicants respectfully submit that the dialog manager of Papineni et al.,

although they teach dynamically switching between a machine-initiative mode and a mixed-initiative mode, still require the maintenance and modification of a list of currently admissible forms being implemented in such that the dialog mode is dynamically adapted. In other words, the dialog manager taught in column 7 is still limited to a form-based dialog manager and is expressly not a state-based model. Thus, there can be no receiving a context changing input associated with speech from a user that changes the dialog context while the spoken dialog is being controlled by the first flow controller that is a finite state model. The steps of comparing the context changing input to at least one context shift is not taught as well as if any of the context shifts are activated by the comparing step, then passing control to an invoked second flow controller (that is a finite state model) indicated by the context shift. Inasmuch as there is no interaction with a finite state-based flow controller taught in columns 6 and 7 of Papineni et al., Applicants submit that there are several limitations that are simply not taught or suggested in the reference. Further, the last step of claim 9 requires that if no context shift is activated by the comparing step, then the method includes maintaining control of the spoken dialog with the first flow controller. Again, since there is no finite-state model that is a first flow controller taught in column 7, Applicants submit that this limitation of claim 9 is also not taught in the reference.

Applicants finally traverses the characterization on page 8 of the Office Action of this last step just discussed above and notes that it cannot be “inherent in the normal operation of the dialog manager as disclosed by Papineni” because Papineni et al. expressly distance themselves from a dialog manager that is state-based and thus, this particular aspect of claim 9 certainly cannot be inherent in the normal operation of their dialog manager.

Accordingly, Applicants submit that there are numerous limitations of claim 9 that are not taught or suggested in the reference and therefore claim 9 is patentable and in condition for allowance. Claims 10-14 depend from claim 9 and each recite further limitations therefrom.

Accordingly, Applicants submit that these claims are patentable and in condition for allowance. Claim 16 is patentable for the same reasons set forth above relative to claim 9. Accordingly, Applicants submit that claim 16 and its dependent claims 17-21 are patentable and in condition for allowance. Claim 23 has been amended in a similar manner to claim 1. Claim 23 is patentable for the same reasons set forth above according to claim 1. Accordingly, Applicants submit that claim 23 and its dependent claims 24-30 are patentable and in condition for allowance.

Rejection of Claims 31-35 Under 35 U.S.C. §102(e)

The Office Action rejects claims 31-35 under 35 U.S.C. §102(e) as being anticipated by Werner (U.S. Publication No. 2003/0171925) ("Werner"). Applicants have cancelled claims 31-35 without prejudice or disclaimer, thus rendering this rejection moot. Applicants anticipate pursuing these claims in a divisional application.

Rejection of Claims 4 and 26 Under 35 U.S.C. §103(a)

The Office Action rejects claims 4 and 26 under 35 U.S.C. §103(a) as being unpatentable over Papineni in view of Brown et al. (U.S. Patent No. 5,699,456) ("Brown et al."). Applicants do not concede that it would be obvious to combine these references, but submit that claims 4 and 26, dependent on an allowable parent claim, are also allowable.

Rejection of Claims 5 and 27 Under 35 U.S.C. §103(a)

The Office Action rejects claims 5 and 27 under 35 U.S.C. §103(a) as being unpatentable over Papineni in view of Abella et al. (U.S. Patent No. 6,044,347) ("Abella et al."). Applicants do not concede that it would be obvious to combine these references, but submit that these claims are patentable inasmuch as claims 5 and 27 each depend from an allowable parent claim.

Rejection of Claims 9-15 Under 35 U.S.C. §103(a)

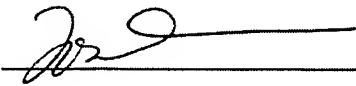
The Office Action rejects claims 9-15 under 35 U.S.C. §103(a) as being unpatentable over Papineni. Applicants traverse the Section 103 analysis and submit that based on the discussion above that substantial evidence supports Applicants position that claims 9-15 are patentable. Certainly, Papineni et al. fails to render obvious each of the first flow controller and second flow controller being each a finite state model.

CONCLUSION

Having addressed all rejections and objections, Applicants respectfully submit that the subject application is in condition for allowance and a Notice to that effect is earnestly solicited. If necessary, the Commissioner for Patents is authorized to charge or credit the **Novak, Druce & Quigg, LLP, Account No. 14-1437** for any deficiency or overpayment.

Respectfully submitted,

Date: November 16, 2007

By: 

Correspondence Address:

Thomas A. Restaino
Reg. No. 33,444
AT&T Corp.
Room 2A-207
One AT&T Way
Bedminster, NJ 07921

Thomas M. Isaacson

Attorney for Applicants
Reg. No. 44,166
Phone: 410-286-9405
Fax No.: 410-510-1433